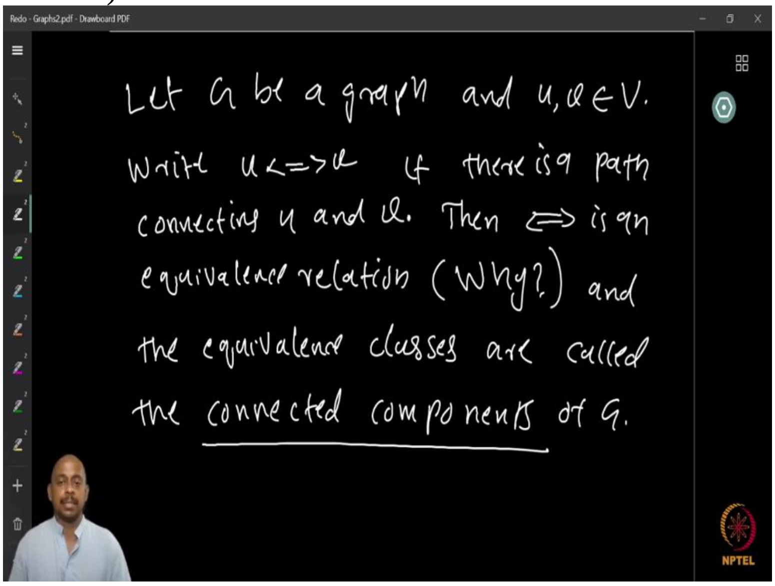
Combinatorics
Professor Doctor Narayanan N
Department of Mathematics
Indian Institute of Technology Madras
Lecture 34
Components, Connectivity, Bipartite Graphs

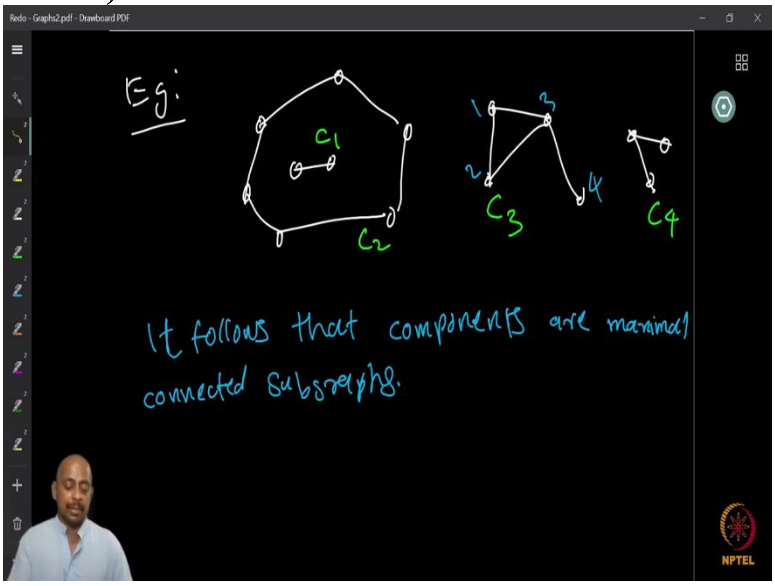
(Refer Slide Time: 00:15)

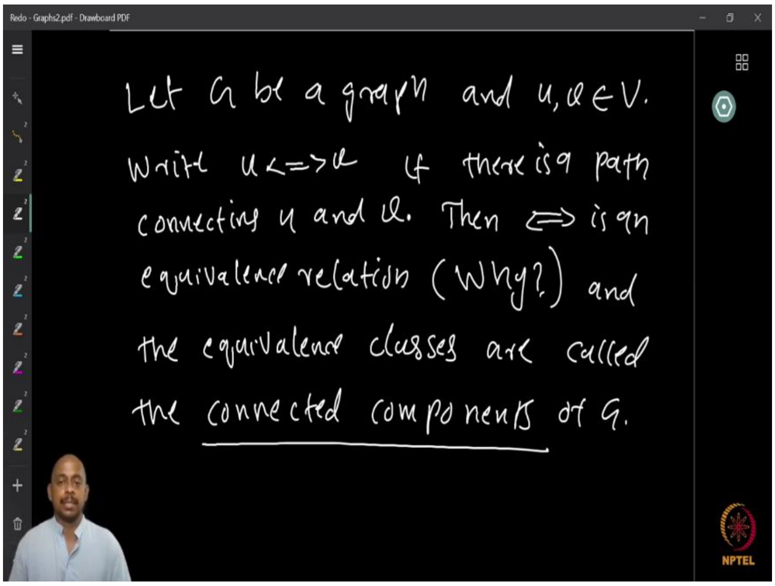


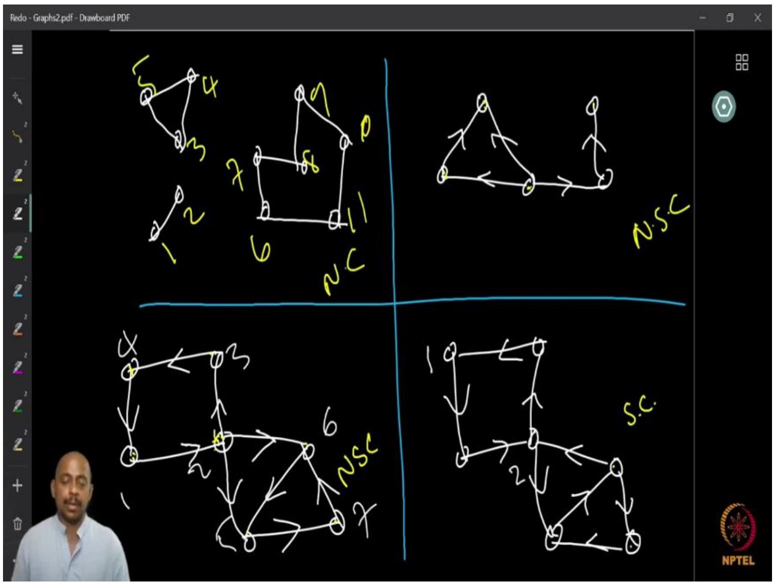
Given a graph and a pair of vertices u and v, we say u is reachable from v and v is reachable from u or u and v we are interconnected if there is a path connecting u and v. If I can go from u to v as well as from v to u then I would say u ⟺ v is an interconnection relation. This interconnection relation ⟺, is an equivalence relation.

I want you to prove this is an equivalence relation showing that it has all the three properties. And then the equivalence classes under the equivalence relations are called the connected components of a graph. For a graph or a digraph, the interconnection relation basically partitions into connected components.

(Refer Slide Time: 01:21)



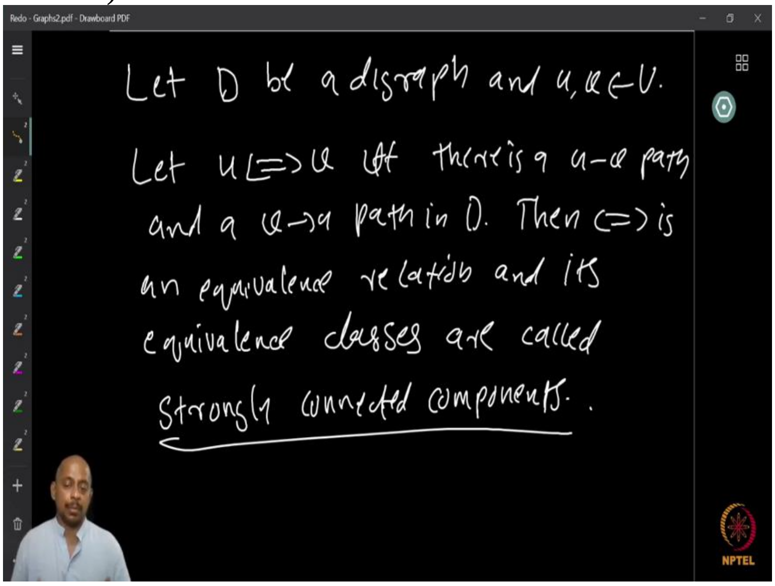




Components are, for example in the previous example, you will see that 1 2 is a component 3 4 5 is another component then 6 7 8 9 and 11 is another component. So, these are the components in the case of graphs. The equivalence classes under this relation is called connected component. Here is another example you have C1 is a component, C2 is another component C3 is a component and C4 is a component. It follows that the components are actually maximal connected subgraphs by the definition, because it is the equivalent classes under the equivalence relation of reachability.

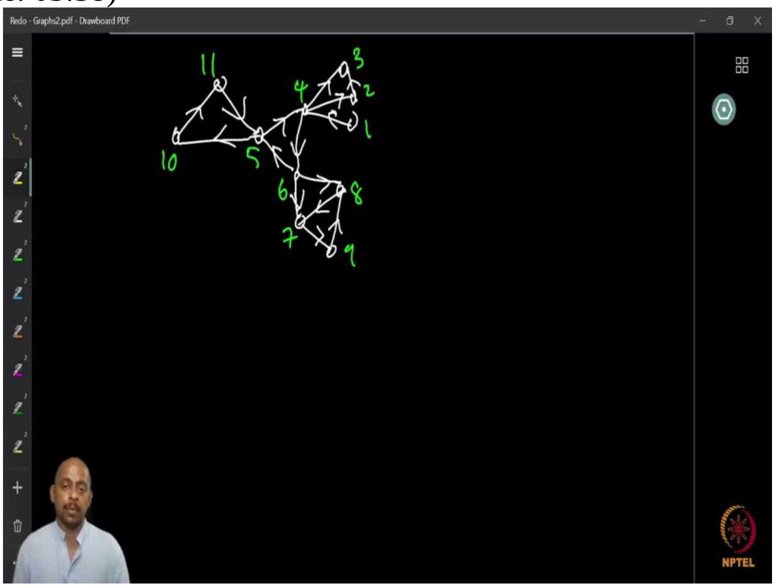
So, u and v are reachable or interconnected then you are looking at the equivalence class which means that it is a maximal connected subgraph. What I wanted to emphasize is that 1 2 3 here is, for example, a connected subgraph but it is not a component because it is not maximal because 4 is also reachable from any of these and similarly any of these are reachable from 4. So, therefore all these 4 vertices for my components where any subset of that is not a component. Similarly, C1 is the component by itself, C2 is the component by itself, C4 is the component by itself.

(Refer Slide Time: 03:12)



D can be a digraph and then again, the interconnection relation u ⟺ v , if and only if, there is a u-v path and a v-u path. Then the interconnection relation is an equivalence relation and again the equivalence classes are called strongly connected components. So, the strongly connected components are the equivalence class under the relation for the directed graphs.

(Refer Slide Time: 03:38)

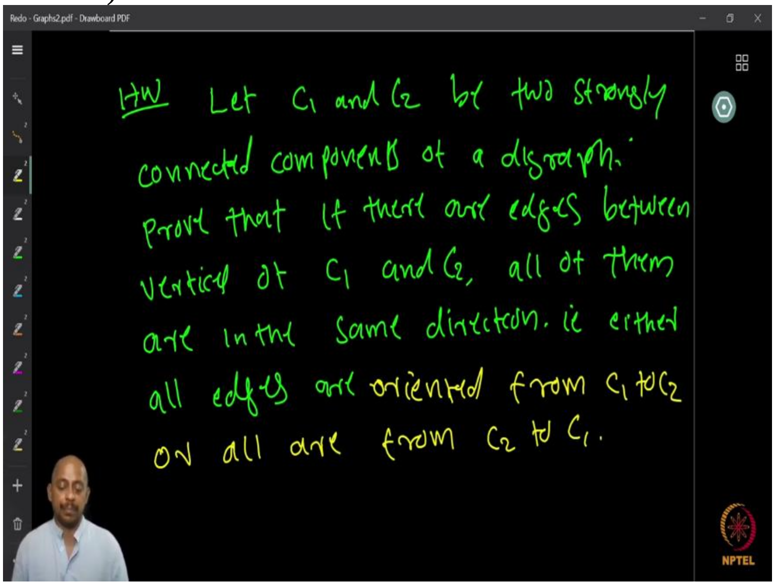


Now, if you look at this, can you find out the strongly connected components? Think about this for a minute before you look at the solution. It is fairly easy you can verify that this part including the vertices 4 5 6 10 and 11 forms a strongly connected component, because you can go from any vertex to any vertex.

For example, 10 to 5 I can go through 11, 10 to 4 I can go from, I am going to 5 and then going to 4. Similarly, to 6 I can go to 4 and then to 6, similarly, if 6 to 4 I can go, 6 to 5 I can go and since, I can go from 5 to any other vertex this also allows us to make sure that this is a strongly connected component.

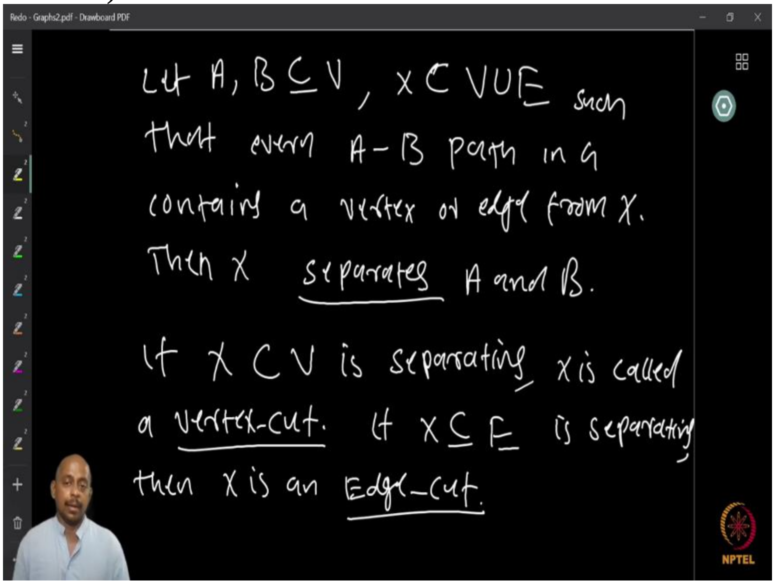
Now, that is because for example I cannot go from 8 to 6, therefore this part will not be there. This is one component then you have another component 7 8 and 9, you can see that it is a directed cycle and there it is reachable between any two. Then one by itself is a component because I cannot reach anywhere from 1, and then 2 by itself is a component 3 by itself is a component. Why 2 itself? Because 2 to 3 I can go but I cannot go back from 3 to 2 and 3 again is itself a component I cannot go anywhere from 3. We have this strongly connected components that we have 1 2 3 4 5 of them.

(Refer Slide Time: 05:39)



Now, a nice homework is to show the following that if C1 and C2 are two strongly connected components then if there are edges between the vertices of C1 and C2 then they must all be in the same direction. Either all the edges are oriented from C1 to C2 that its starting point is in C1 and ending point in C2 or all the edges are starting at vertices in C2 and ends and the vertices of the component C1.

(Refer Slide Time: 06:15)

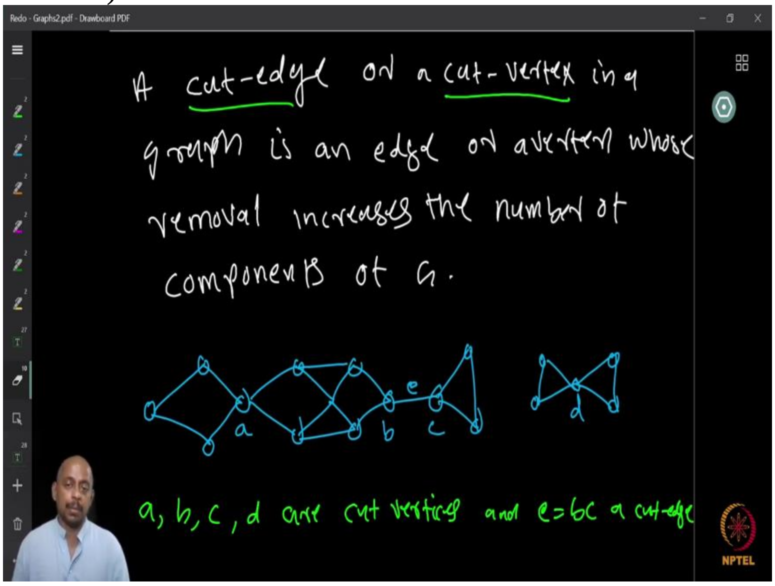


Now, a couple of more definitions. If A and B are subsets of the vertex at V, X ⊂ V ∪ E, so X is some vertices and edges. Now, if A B and X have this property that every A-B path in the graph G contains a vertex or edge from the set X.

Then if I remove X then A and B are basically disconnected. Then we can call X to be a separating set. So, X separates A and B, if X is a subset of V ∪ E such that every A-B path in G contains a vertex or edge from X. If I X is a subset of V alone, then this is called a vertex cut.

A separating vertex set is called a vertex-cut, similarly as separating edge set if X is subset of E alone there are no vertices in X, but just edges then it is an edge-cut, you remove the edges of X I can destroy the connection between A and B. It is basically an A B edge-cut and similarly A B vertex cut.

(Refer Slide Time: 07:46)

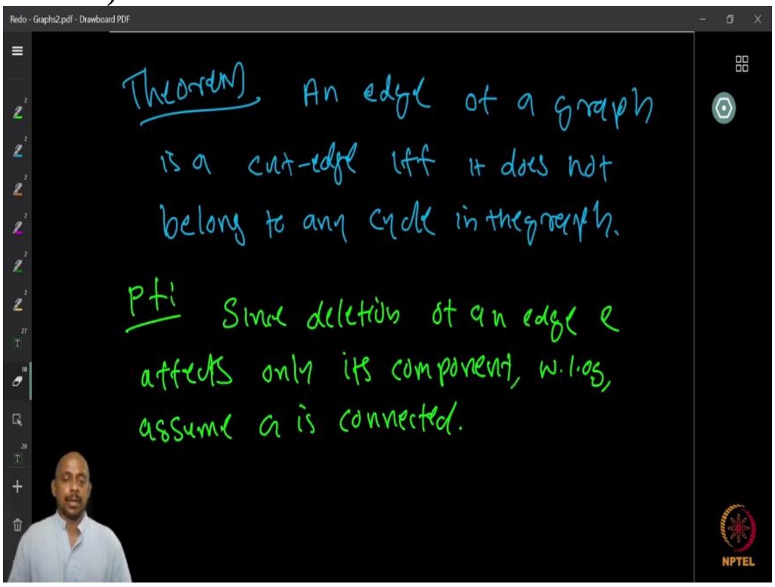


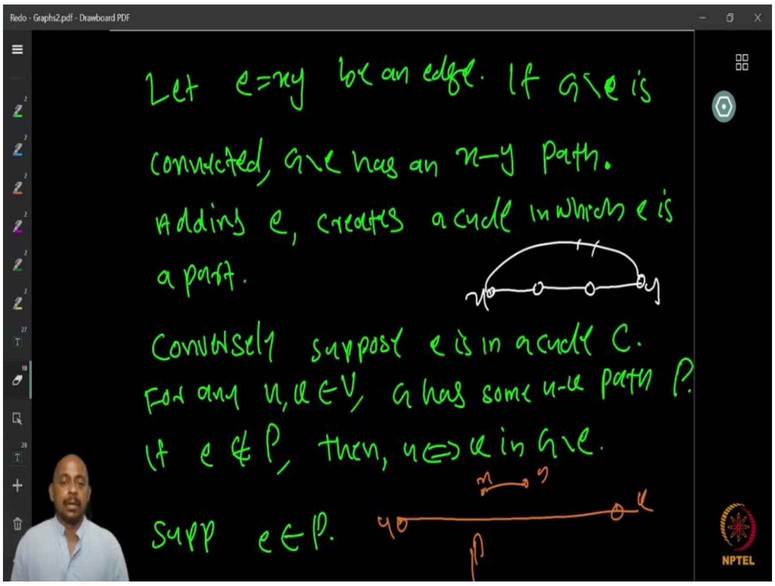
A cut-edge or a cut-vertex in a graph is either an edge or a vertex whose removal increases the number of components of G. Or one can modify it slightly by saying that either it increases the number of components or it gives you the trivial vertex, in some cases that could be useful, but for the time being we will just assume this.

A cut-edge or a cut-vertex is basically an edge or a vertex whose removal increases the number of components of G. Here are some examples; if I look at the entire graph then ‘a’ is a cut-vertex. Why is that? Because if I remove ‘a’ then the number of components in the graph increases from 2 to 3. Then there are 3 components now, and therefore a was a cut-vertex.

I just removed a of course it also removes the edges incident to a. Similarly, I can see that b is a cut-vertex, c is a cut-vertex and d is also a cut-vertex. This disconnects therefor, c is a cut-vertex, similarly d is cut-vertex we have several cut-vertices. What about the cut edges? e is the only cut-edge here, if I remove e, it increases the number of components and there is no other edge whose removal increase the number of components. It is just by removing one edge you can increase the number of components. So, e is a cut-edge and a b c d are cut-vertices of this graph.

(Refer Slide Time: 09:44)





Now, If you think about the cut edges you can show the following theorem, an edge of a graph is a cut-edge if and only if it does not belong to any cycle in the graph. Can you think of a similar result for cut-vertex also? Can you say something about the cut-vertices, not exactly the same but think of this.

Here we are going to prove this theorem that an edge is a cut-edge if and only if does not belong to a cycle in the graph. Usually a cut-edge is also called a bridge; we also have this notion bridge. If you see that e is a bridge then, you can assume that it is basically an edge whose removal increases the number of components.

The deletion of an edge affects only its component, to prove this theorem you notice that if I have several components in the graph to start with and if I delete an edge the number of components in the graph increase only if the components will itself increases the number of components because this edge cannot change the connectivity of any other component.

Therefore, without loss of generality we can assume G is connected because if it increases the number of components then it must definitely increase the number of components from its part its own component. We start the assumption that our graph is connected.

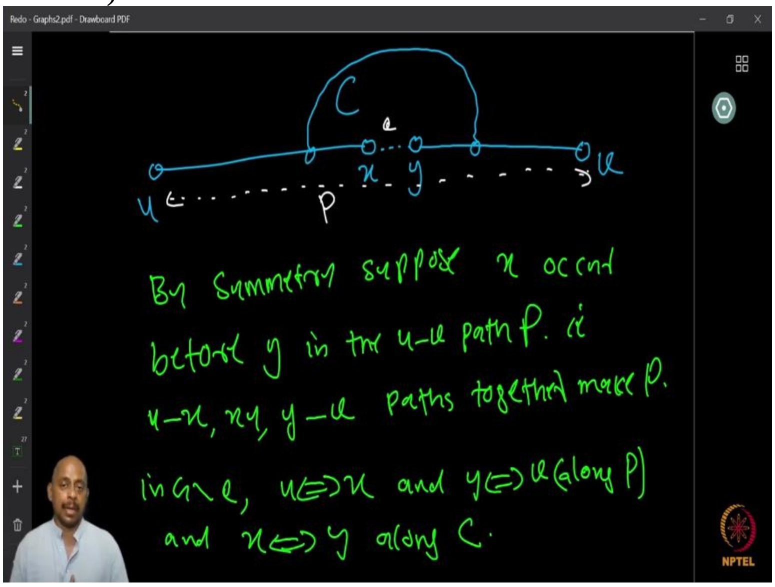
Now, first part, to prove one direction is very easy, if you take e to be any edge let us say e = xy then G\e is connected means that G\e has an x-y path because by a graph being connected we say that there is a path between any two vertices. G\xy is connected means that x to y there must be a path, now x to y if there is a path, take the x-y path and in the graph we have there is edge e.

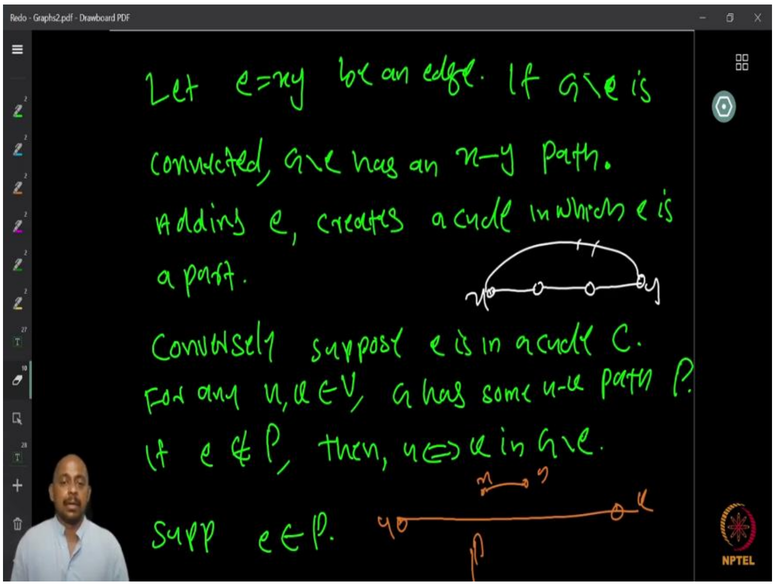
Now, the edge e basically connects x to y if I have an x-y path then the edge e allows us to go from y to x. Now, the definition of a cycle is that you take a u-v path and then take the edge v to u, that is the definition of a cycle. Therefore, from that you will see that the graph must contain a cycle.

If removal of an edge keeps the graph connected then we know that this edge is part of a cycle. Now, we are going to prove the converse. To prove the converse, we assume that e is in some cycle C. We have to show that after removing the edge e the entire graph remains connected.

For any pair of vertices u and v, we have to show that we have some u-v path. If you start with the graph G, we are looking at a graph which is connected because without loss of the generality we can assume that. Therefore, since G is connected; G has a u-v path let us call this path as P. Now, if the edge e that we are going to remove is not part of the path P then removing this will not change the connectivity of u and v, it will still be connected, u to v will be still connected, there is a path from u to v in G\e. So, u is interconnected to v in G\e if e is not a part of the path P. Suppose e is part of the path P.

(Refer Slide Time: 13:45)





If e is actually part of the path P, then we can assume that by symmetry, you have this u, v for any arbitrary pair of vertices you are taking. So u to v there is a path and u and v are connected or not in the remaining graph is what we want to check. So, in the previous example we said that if e is not part of the u-v path that we are considered then u and v are not going to be disconnected after removing, because the path P itself is still there.

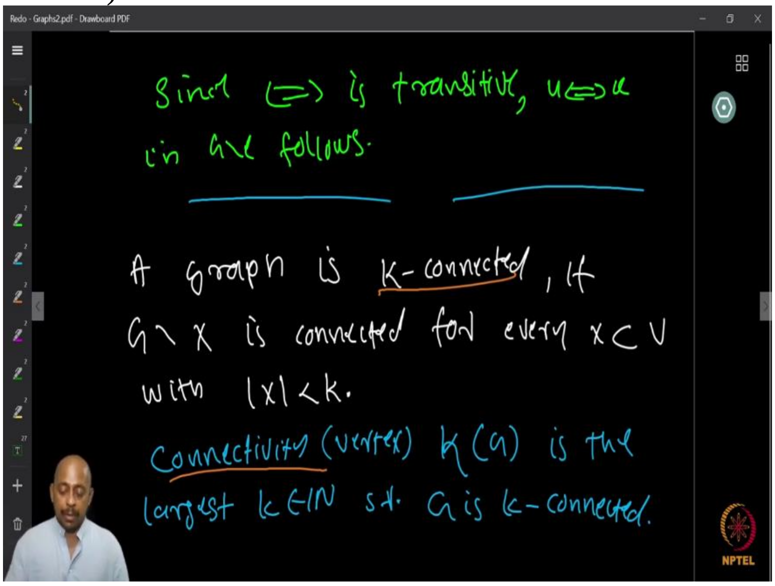
Our assumption is that the edge e is part of the path, then if I remove what happens. If e is part of the path then the u-v path must go through the edge e and then I have a u to x and x to y edge and then y to v path. Or it can be, u to y then y to x and x to v, but because of symmetry I can assume one of these. So, I can say that in the u-v path the edge xy, x appears before y in the path.

Now, with this assumption you have the property that u to x there is a path u x interconnection is there then (x, y) the edge is there, and then y to v the path is there. The three interconnection relations by transitivity will give you u to v is connected. But now we have removed the edge (x, y), what happens if I remove (x, y)? If I remove (x, y), we said that the edge e was a part of a cycle, because it was part of cycle x to y there must be another path or a walk so there is an x-y walk and together with the (x, y) edge with this path, x-y path it was forming a cycle.

But now if I remove the edge (x, y), still x to y there is this path, because it was part of the cycle and then we added, now x and y are interconnected through this path in C. So if C is the cycle, then through the remaining edges of C there is a path connecting x to y, so x and y are interconnected. Any walk is basically saying that there is also a path.

Now, u to x there is a reachability. So, u to x are reachable then, x to y reachable through the path in C then y to v is reachable through the original path P it is a sub path of the path. So, u to x the path through the original path, y to v the original path is still there then x to y there is a path through the cycle C that were xy was part. Now, the interconnection relation is an equivalence relation. Therefore, because it is transitive if u-x there is a connection, and x-y there is a connection and y to v there is a connection then by transitivity u to v there is a connection.

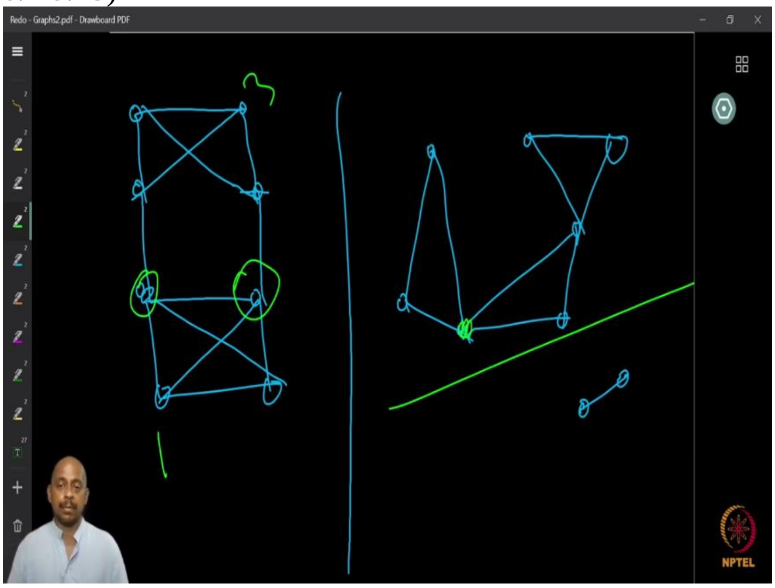
(Refer Slide Time: 17:10)



Therefore, there must be reachability between u to v and therefore, G\e is also connected. This is the completion of the proof. Now, a graph G is defined to be k-connected if I cannot remove less than k vertices to disconnect the graph. So, I start with the graph G, which is connected let us say and if I cannot remove less than k vertices to make the graph disconnected then the graph is said to be k- connected.

The vertex connectivity of the graph is the largest k such that G is k-connected. Let $\kappa(G)$ be the largest k such that G is k-connected. If I take $\kappa + 1$ and then there is a subset of vertices whose removal disconnects the graph. Therefore, the connectivity is the largest k such that, G\X is connected.

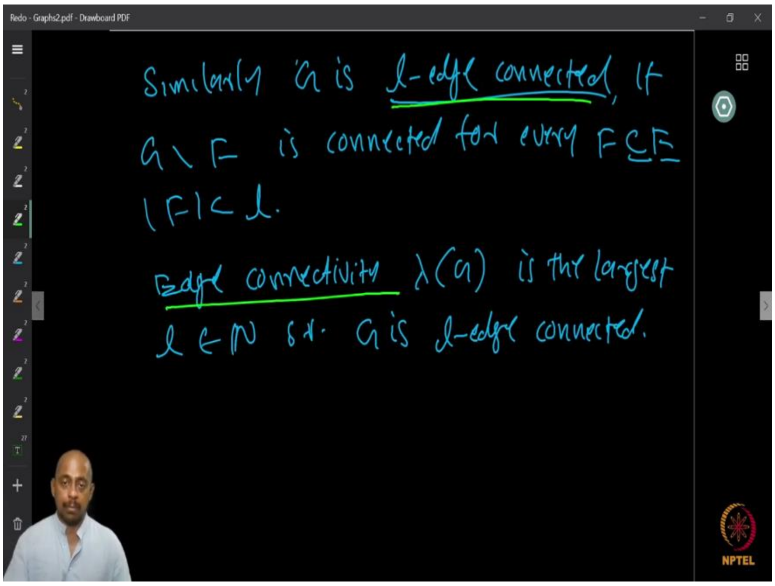
(Refer Slide Time: 18:45)



Here are some examples, if you look at this graph, it is 2-connected because one can verify that there is no cut vertex in the first part. We do not have any cut vertex here. On the other hand it is 2 connected and its connectivity is two because I can find a separating set of size 2. For example, this vertex and this vertex, if I remove these 2 then there is no path from let us say one to three therefore the number of components increases. Therefore, this is the separating set and then you have to remove 2 vertices to make the graph disconnected and with 2 you can actually disconnect it and therefore connectivity is actually 2.

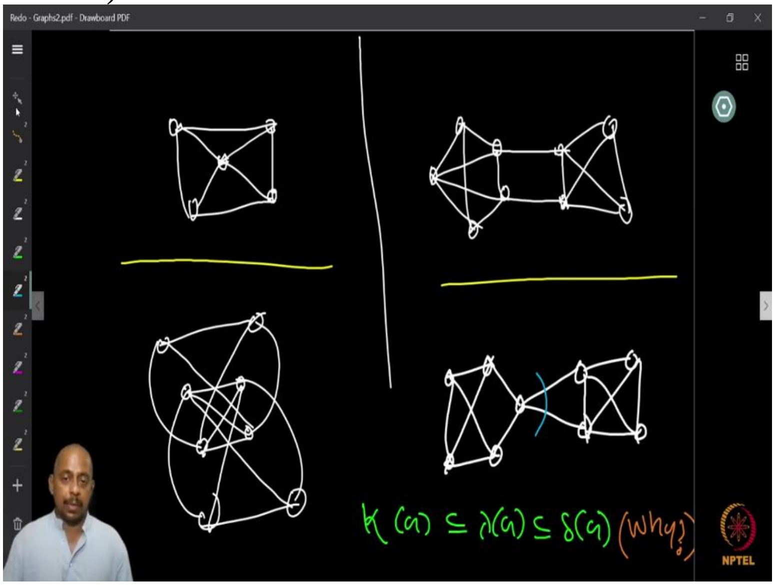
On the other hand, this graph has connectivity 0, because the entire graph is not connected. Even removing zero vertices the graph is still disconnected. On the other hand, if I just look at one part of this let us say I am just looking at this as a graph. Then this graph has connectivity one, because it has cut vertices. This is a cut vertex, so on removal disconnects the graph. It increases the number of components and the graph by itself is connected.

(Refer Slide Time: 20:14)



Now, a graph G is l-edge connected if G\F is connected for every F subset of E. So, similar to vertex connectivity, I can talk about edge connectivity we say a graph is l-edge connected if G\F is connected for every F subset of edges, where the cardinality is strictly less than l. By removing less than l, I cannot disconnect the graph. Therefore, it is at least l. Its connectivity is the largest again, the largest natural number such that G is l-edge connected.

(Refer Slide Time: 21:01)



Here is another set of examples, if you look at this graph what is the edge connectivity? If you look at the first graph it is 3-edge connected because we know it I cannot remove just 2 edges to make the graph disconnected, this part. But I can remove just three edges to make it disconnected therefore it is a 3-edge connected graph.

On the other hand the graph on the right side is 2-edge connected because I cannot remove just one edge, there is no bridge but I have to remove 2 edges, I can remove 2 edges to make the graph disconnected. These 2 for example. So therefore its edge connectivity is 2 and this connectivity for the previous graph was 3.

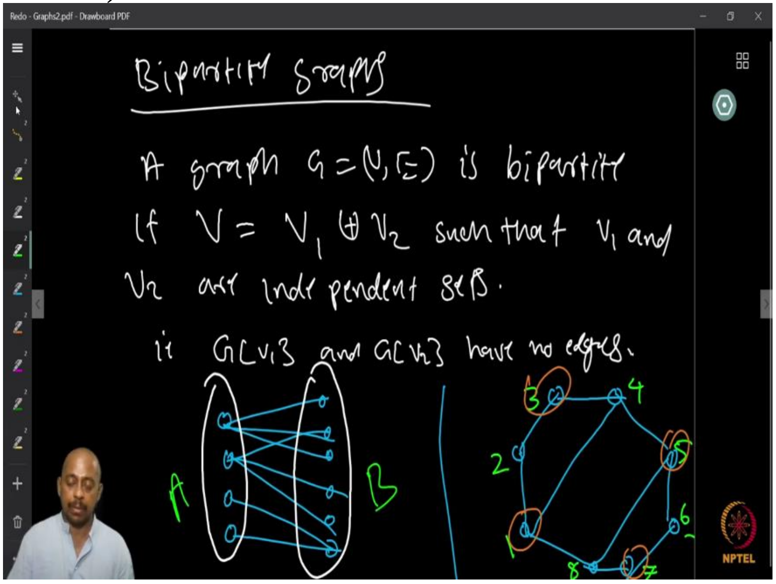
Now, what about the edge connectivity of the graph 3 here? What about this? If you look at the edge connectivity of this, it is not immediately clear what is it but if you look through it carefully you will see that it is actually 2, you can find a 2 edges whose removal disconnects graph.

So, what are these 2? You should look for it before you continue, if I remove these 2 cross edges then you will see that the graph is actually disconnected, this part there are these 2 parts which are superimposed and there are no edges connecting let us say the blue vertices to the other remaining vertices. So, the graph is 2 connected.

Then what about this one? Here the vertex connectivity is actually one, you can verify that vertex connectivity is one because it is a cut-vertex but the edge connectivity for example is 2 because I can remove these 2 edges to disconnect the graph, either these 2 or the these 2. Either of these will disconnect the graph.

So, you can verify this and then look at several examples. I encourage you to look through several examples before continuing further then you will observe the following that for any graph the vertex connectivity $\kappa(G)$ is upper bounded by the edge connectivity $\lambda(G)$ and then this is upper bounded by the minimum degree of the graph. This should be kind of intuitively clear but I want you to give a formal proof. So, think about this. Why vertex connectivity is less than or equal to edge connectivity less than or equal to minimum degree and we will use this result later.

(Refer Slide Time: 25:03)

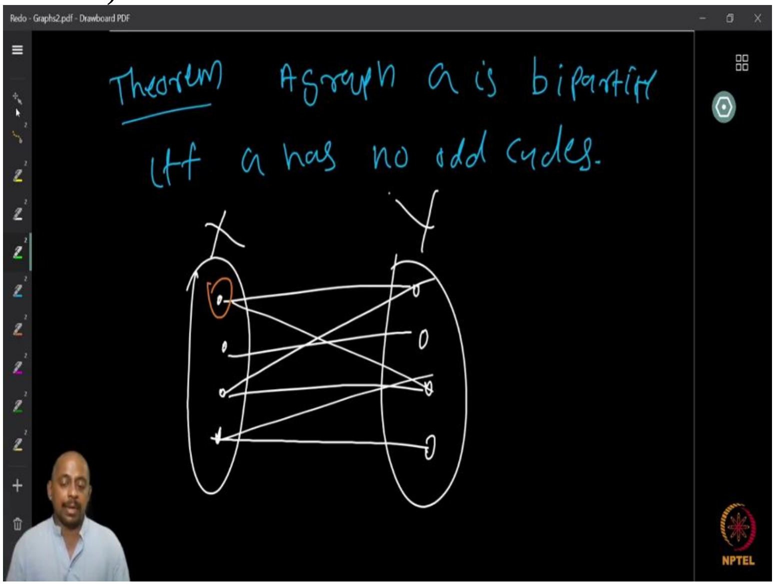


Here is another important notion of bipartite graphs. So, the graph G is bipartite if I can write the vertex set as a disjoint union of 2 sets V_1 and V_2. Let us say that both V_1 and V_2 are independent sets, if you remember what are independent sets? An independent set is a subset of vertices where there are no edges between these vertices, then it is an independence set.

So, if I can partition the entire graph to 2 parts, 2 independent sets and all the edges are basically across, so it goes from one of the sets to the other but these 2 are independent there are no edges. Here is a nice representation of the bipartite graph, I have this part let us say A and there is this part B. A and B are independent sets and the edges are between A and B. Any edge has one of the endpoints in A and the other endpoint in B.

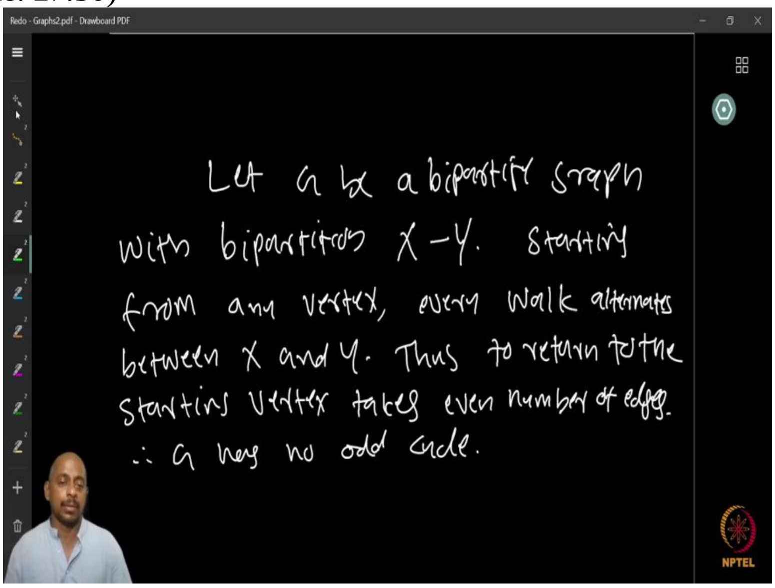
The induced subgraphs on V_1 and V_2 have no edges then it is a independent set. An example on the right hand side, so the vertices marked in red form an independent set, there are no edges between them, and the remaining vertices from another independent set therefore the graph is bipartite, you can verify that all the edges are going from the red vertices to the other vertices.

(Refer Slide Time: 26:42)



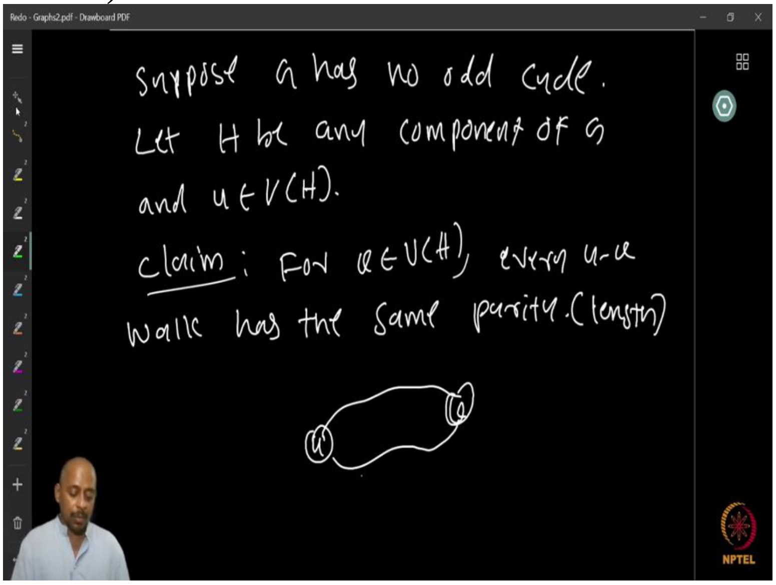
Now a graph, this is a very important theorem; a graph is bipartite if and only if the graph has no odd cycles. So, this theorem I want you to try to think to prove, we will prove it here, but just think about this for some time and one part is immediate so I can see that if the graph is bipartite then it cannot have odd cycles, why is that? Because if the graph is bipartite let us think of any cycle. I start from a vertex then after going and taking edges I have to come back to that vertex. Now, since, the graph is bipartite if I take any edge it actually alternates the sides.

(Refer Slide Time: 27:30)



Therefore, since, any walk alternates between x and y to return to the starting vertex you have to take an even number of edges and because every closed walk is of even length there is no odd cycle, because if the old cycle itself is a closed odd walk. Therefore, this is clear.

(Refer Slide Time: 27:57)



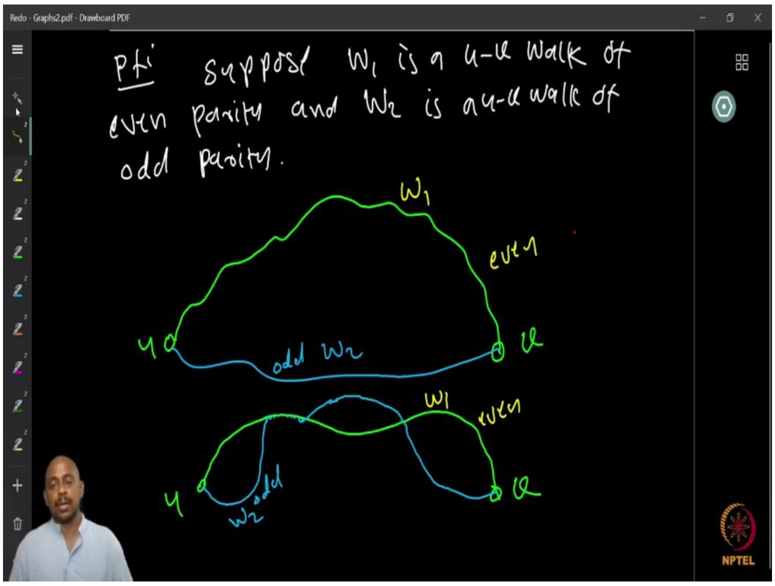
Now, to prove the other side, suppose the graph has no odd cycle we want to construct a bipartition for the graphs. We are going to prove graph is bipartite by constructing a bipartition. Now, let us take H to be any component of the graph, because if the graph is bipartite then every component is bipartite because I have a partition into 2 independent sets then look at the components each one has a partition into 2 independent sets.

Therefore, I just start with one of the components let H be any component of G and take a vertex u in H. What I can do is that, for the vertex u in H every u-v walk, so take the starting vertex u then fix any other vertex v in the component then look at every u-v walk in the graph, because if I take any vertex outside the component there is no walk at all.

So, I can look at the u-v walks from any vertex u to the vertex v. Once you fix a vertex v also, then you can have several u-v walks, in the graph there could be several paths or walks between a pair of vertices. Look at every u-v walk my claim is that every u-v walk has the same parity. Which means that the length of every u-v walk is either all odd or all of the walks are even length.

Why is this true? Our assumption is that we start with a graph which has no odd cycle then we take some component then I fix some vertex in this component then I look at all the u-v walk for every vertex v, and then find the all u-v walk. Once you fix v every u-v walk has the same parity it is all odd length or even length. Why is that? Now, suppose you have two different u-v walk with different parities, so u to v there is an odd walk and then u to v there is suppose an even walk.

(Refer Slide Time: 30:29)



So, now let us say that W_1 is a u-v walk with even parity, and W_2 is a u-v walk of odd parity. The walks may be intersecting, may not be intersecting, we do not care about they are intersecting or not, because when we are talking about walks, walk itself can intersect itself. Let us look at all possible u-v walks, I have a walk W_1 from u to v then there is an odd walk u to v which is W_2 , one is even and one is odd.

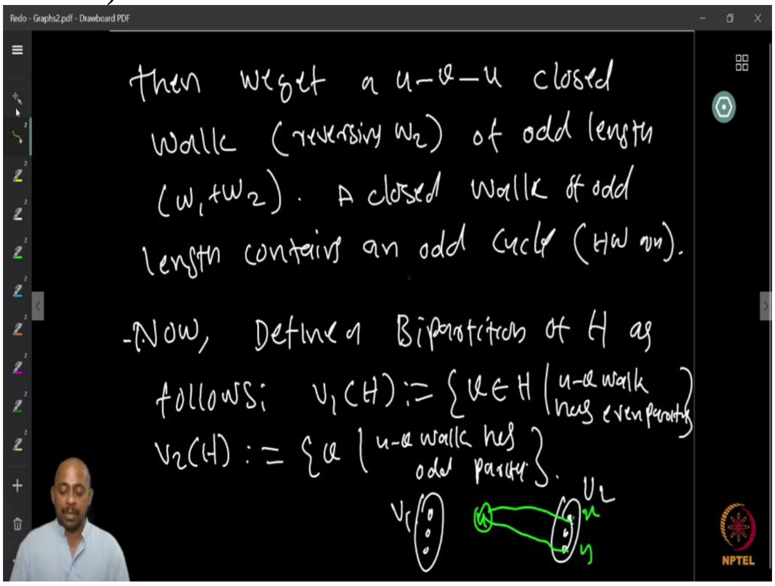
Similarly, it would be like intersecting. $W_1$is an even walk then W_2 is an odd walk. No matter what I look at the u to v walk and then v to u walk through W_2. I take W_1 first then take W_2 if I go from u to v through W_1 then v to u through W_2 or in this case it would be repeating vertices, edges sub-paths, everything does not matter.

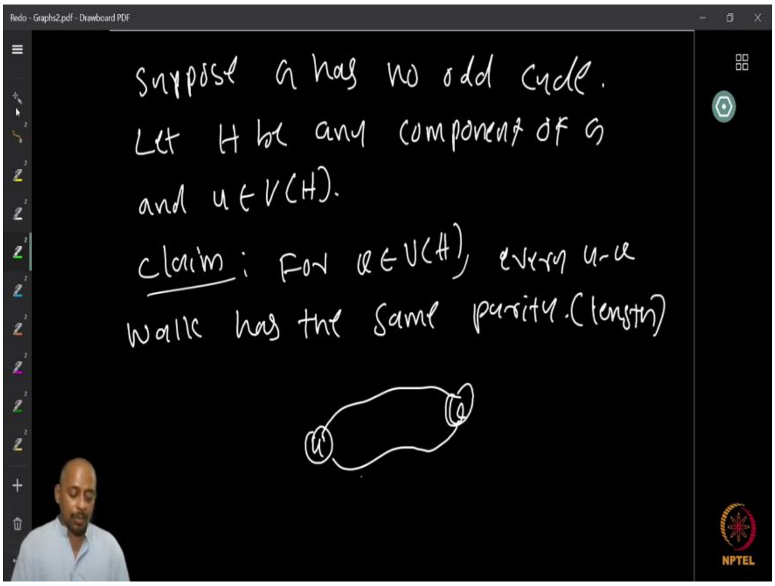
Whatever it is I look at this walk from u to u itself now this walk is a join of 2 walks W_1 and W_2. So the length of this walk is the sum of the lengths it is basically an odd length closed walk, I get a closed walk of odd length.

We proved, at least as a homework I gave you, to prove that if there is a closed walk of odd length then it must contain an odd cycle. A closed walk of even length does not tell you there is an even cycle but there is a closed walk of odd length it tells you there is an odd cycle.

I want you to come up with an example of a graph with some closed even length walk and show that this walk does not contain a cycle as a subgraph, but you can have of course cycles but you have examples without cycles, so that is what I want you to do and once you have the property that every closed walk of odd length contains an odd cycle, you will see that the graph must contain an odd cycle, which is a contradiction.

(Refer Slide Time: 33:05)





The u to v, v to u closed walk which is the walk W_1 plus W_2 contains an odd cycle. Once you have this, we know that every walk must have the same parity. So, once you have u and every fix at v, u-v walks all must have the same parity. Since all of them have the same parity, I can say that if u-v walk is of even length I collect such v, u-v walk is of odd length, I also collect such v.

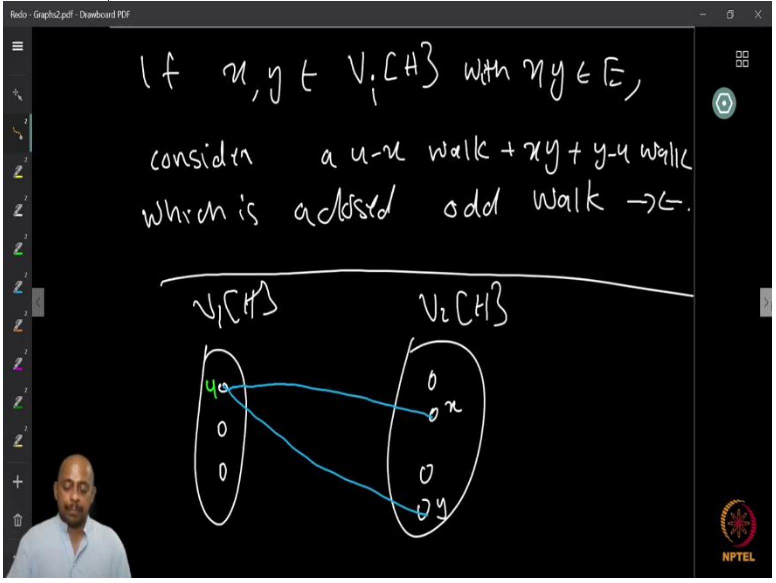
Pick out the vertices which has old walks from u and pick out the other vertices which has even walks from u, because the graph is connected, the component H is connected. Every vertex other than u there is a walk from u, including u itself, u to every other vertex there is a walk either of odd length or even.

So, collect them together and then form the subgraph as follows. So, the partition is as follows, V_1 of the H is defined as set of all vertices in H as that u-v walk has even parity then V_2 of H is the set of all vertices such that u-v walk has odd parity. Claim is that V_1 and V_2 forms a partition of H into independent sets. V_1 is independent and V_2 is independent. Now, why is this? We have to prove that V_1 is independent and V_2 is independent.

Let us take one of them, so that there cannot be an edge between any 2 vertices. So, if x and y are vertices in this V_2 let us say and xy is an edge. If xy is an edge I claim that the graph contains an odd cycle. Why is that, because if xy is an edge, u to x there is some walk,which has let us say odd parity.

Then u to y there is a walk which has also odd priority which means that there sum has exactly even number of edges. Together with the edge y you can get u to x, x to y and y to u is a closed walk of odd length which gives an odd walk and then there is an odd cycle. Similarly, if u-x is even then y-u is also even because I collected all the vertices with the same parity to v. Therefore, even plus even plus one is again odd therefore I again get odd cycle.

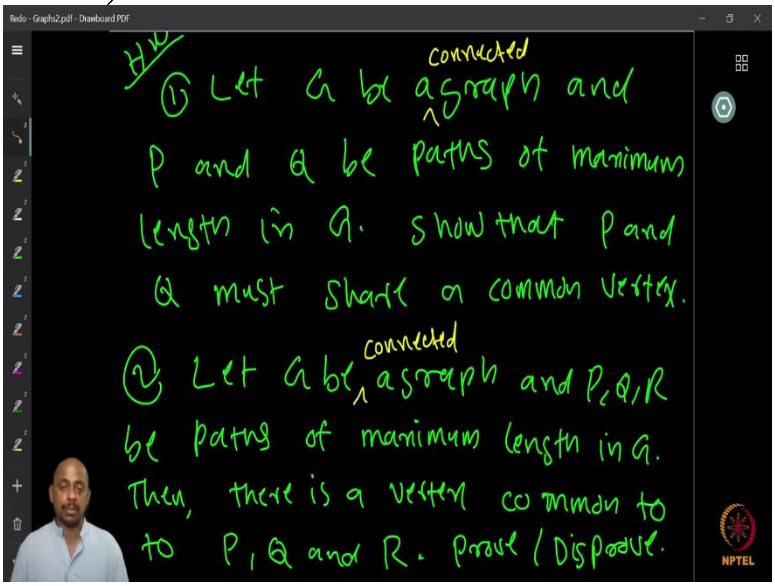
(Refer Slide Time: 35:57)



Therefore, we get the fact that V_1(H) and V_2(H) are independent therefore they are both independent sets and therefore it is a bipartition. For each component I can get a bipartition using this, if there is no odd cycle in each component I can get a bipartition, now if I have bipartitions of several components.

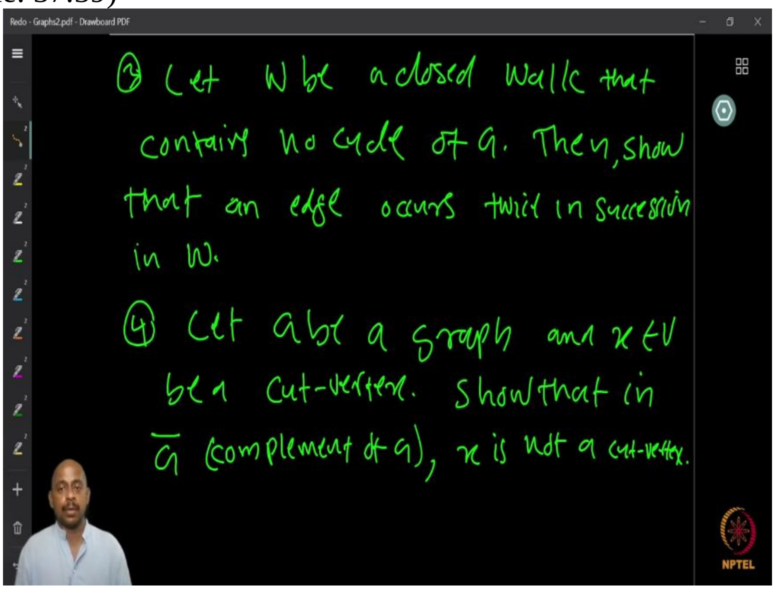
I can put them together whichever way I want, V_1 of the first graph, V_2 the second graph again V_1 of the second graph, V_1 of the first graph V_2 of the first graph, V_1 of the second graph V_2 of the first graph, etc. you have the partition and that will give you the bipartition of the entire graph.

(Refer Slide Time: 36:45)



Here are some homework questions, let G be a connected graph and P and Q are paths of maximum length in G. Show that P and Q must share a common vertex. This is a nice homework. Then if G is a connected graph and P, Q, R are paths of maximum length, can you show that there is a vertex common to all the three paths P, Q and R. Either prove or disprove, but I want to emphasize that this is not an easy problem, the second question. If you do not get an answer, do not worry about it, but try to think of this for some time. It is very interesting questions if you get an answer, please feel free to let me know.

(Refer Slide Time: 37:39)



The question three, let W be a closed walk that has no cycle of the graph then show that an edge occurs twice in succession in W. So, suppose we have a closed walk which does not contain any cycle of the graph then some edge must occur twice in succession in W, for example, I mean it will be reversing, I go from u to w then w to u back this must happen for at least one edge, if closed walk does not contain any cycle.

Then the 4th question is that if G is a graph and x is a cut vertex then show that in the complement of the graph, the complement of a graph is the graph obtained by making all the edges as non-edges and all the non-edges as edges. So, in this complement of the graph, x is not a cut vertex. So, these 4 questions I want you to think about and try to solve and with that let us stop for today.